All-Source Track and Identity Fusion

June 2000

Stefano Coraluppi, Craig Carthel, Mark Luettgen, and Susan Lynch
ALPHATECH, Inc.
50 Mall Rd
Burlington, MA 01803, U.S.A.
Tel. +1-781-273-3388 ext. 366
Fax: +1-781-273-9345
Email: stefano.coraluppi@alphatech.com

ABSTRACT

This work addresses the association of moving-target indicator (MTI) tracks, EO and SAR imagery (IMINT) tracks, and signal intelligence (SIGINT) tracks, and the fusion of the corresponding report-level kinematic and identity information.

Our fusion algorithm is based on hypothesis-management logic which recursively processes incoming frames of data from upstream trackers. The logic includes hypothesis generation, scoring, and pruning components. These components are based on track information, kinematic-state information, and vehicle identity information.

Key track-level performance metrics for our fusion algorithm include the probability of correct track-to-track association and track fragmentation. We study the performance of the algorithm with simulated single-sensor tracks, for two scenarios of interest. The first scenario is based on a data collection for a set of 30 GPS-equipped targets, while the second is based on simulated ground truth for a set of 8 scattering targets.

KEY WORDS Sensor Fusion, Multi-Sensor Multi-Target Tracking, Hybrid Systems, Nonlinear Filtering

ACKNOWLEDGEMENT We wish to thank Professors Alan Willsky and John Tsitsiklis of MIT for valuable discussions on this work.

1. INTRODUCTION

Existing algorithms for multi-target tracking are generally limited to single-sensor platforms, or to multi-sensor platforms where the sensors have similar characteristics, e.g. multiple GMTI platforms; similarly, much of the work on track-to-track association algorithms has been limited to trackers with similar characteristics, e.g. multiple GMTI trackers [1]. Relatively little work has been done to address the more general fusion problem where an arbitrary number of trackers with widely varying sensor characteristics are employed. More significantly, to our knowledge our work is the first to do so in a multiple-hypothesis framework, combining hypothesis testing and nonlinear filtering technology to enable accrual of information and deferred resolution of association hypotheses.

Our algorithm takes as input the tracks from an arbitrary number of GMTI, EO and SAR imagery (IMINT), and signal intelligence (SIGINT) trackers, as well as elevation data if this is available. The algorithm fuses

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (FROM - TO)
01-06-2000	Conference Proceedings	xx-xx-2000 to xx-xx-2000

4. TITLE AND SUBTITLE
All-Source Track and Identity Fusion
Unclassified

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Coraluppi, Stefano ;
Carthel, Craig ;
Luettgen, Mark ;
Lynch, Susan ;

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME AND ADDRESS
ALPHATECH, Inc.
50 Mall Rd.
Burlington, MA01803

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS
Director, CECOM RDEC
Night Vision and Electronic Sensors Directorate, Security Team
10221 Burbeck Broad
Ft. Belvoir, VA22060-5806

10. SPONSOR/MONITOR'S ACRONYM(S)
11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APUBLIC RELEASE
,

13. SUPPLEMENTARY NOTES
See Also ADM201258, 2000 MSS Proceedings on CD-ROM, January 2001.

14. ABSTRACT
This work addresses the association of moving-target indicator (MTI) tracks, EO and SAR imagery (IMINT) tracks, and signal intelligence (SIGINT) tracks, and the fusion of the corresponding report-level kinematic and identity information. Our fusion algorithm is based on hypothesis-management logic which recursively processes incoming frames of data from upstream trackers. The logic includes hypothesis generation, scoring, and pruning components. These components are based on track information, kinematic-state information, and vehicle identity information. Key track-level performance metrics for our fusion algorithm include the probability of correct track-to-track association and track fragmentation. We study the performance of the algorithm with simulated single-sensor tracks, for two scenarios of interest. The first scenario is based on a data collection for a set of 30 GPS-equipped targets, while the second is based on simulated ground truth for a set of 8 scattering targets.

15. SUBJECT TERMS
Sensor Fusion; Multi-Sensor Multi-Target Tracking; Hybrid Systems; Nonlinear Filtering

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	Same as Report (SAR)	17	Fenster, Lynn lfenster@dtic.mil
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER International Area Code Area Code Telephone Number 703767-9007 DSN 427-9007

these tracks, and results in a set of tracks that have improved continuity, reduced errors in kinematic state estimates, and improved identity (target type) information. Indeed, the trackers of interest have complementary characteristics: GMTI trackers identify moving targets (or stationary, rotating targets); IMINT and SIGINT trackers are effective at identifying target identity, and potentially identify both stationary and moving targets. A key point is that, while the algorithmic details we describe are specific to these classes of trackers, the framework is quite general, so that other trackers based on sensors with different detection characteristics can be easily accommodated.

This paper is organized as follows. In Section 2, we describe our hybrid-state modeling paradigm for target kinematics and our hybrid-state kinematic filter, which generalize the well-known Extended Kalman Filter (EKF). In Section 3, we motivate our Markov chain model for target identity and describe the identity filtering equations. Models for the upstream trackers are discussed in Section 4. Section 5 describes the framework for the *ATIF* (all-source track and identity fusion) algorithm; the recursive hypothesis-management logic includes hypothesis generation, scoring, and pruning components. Section 6 defines performance metrics of interest and provides two case studies, associated with two scenarios of interest. Concluding remarks that summarize our work and identify areas of continuing work are given in Section 7.

2. KINEMATIC MODELING AND FILTERING

The target attributes that we exploit for tracking purposes are the (time-varying) kinematic state as well as the (static) identity, or vehicle type. In this section, we describe our kinematic modeling and filtering. Further details can be found in [2].

A number of multiple-model and hybrid-state approaches to target tracking with MTI sensors have been proposed [3]. We will explore the use of a hybrid-state model to address our multi-sensor problem. In our hybrid-state model, the discrete-state process is a two-state continuous-time Markov chain. The state at time t is given by $\xi(t) \in \{m, s\}$, where *m* and *s* represent *move* and *stop* states, respectively. Transition rates are given by λ_{ms} and λ_{sm} for *move-stop* and *stop-move* transitions, respectively.

The continuous-state dynamics depend on the discrete motion state. In particular, we have:

$$\dot{X}(t) = F(t)X(t) + w(t,\xi(t)), w(t,\xi(t)) = \begin{bmatrix} 0 & 0 & w_x(t,\xi(t)) & w_y(t,\xi(t)) \end{bmatrix}', \tag{2.1}$$

where we have zero-mean white Gaussian noise with $E\left[w_x^2(t,\xi(t))\right] = q_x 1[\xi(t) = m]$ and $E\left[w_y^2(t,\xi(t))\right] = q_y \delta(t) 1[\xi(t) = m]$. (The characteristic function $1[\cdot]$ equals unity when its argument holds, and equals zero otherwise).

Transitions in the continuous state at discrete-state transition times are as follows. The first two components of the state, which are the vehicle position, are continuous at transition times. The velocity components are discontinuous at discrete-state transition times. In particular, at a *move-stop* transition, the velocity jumps to zero. At a *stop-move* transition, the velocity is modeled as a zero-mean random vector with covariance Σ_T.

In order to use the hybrid-state model in a tractable way, it is necessary to introduce the following fundamental assumption for the dynamics relative to the time sequence $(t_1, t_2, ...)$:

- The hybrid-state dynamics obey a fundamental *minimality* or *sampling* assumption, whereby a target undergoes at most one discrete-state transition between frame times.

Note that a sensor report may not identify the motion state of a vehicle as being *move* or *stop*. Similarly, the lack of a report for a target does not necessarily identify the motion state of the target. However, our

hybrid-state filtering equations are conditioned on a particular discrete-state sequence; thus, we identify an unambiguous sequence of discrete states at each of the times $(t_1, t_2, ...)$, for each track hypothesis.

Given a sequence of discrete motion states at the times $(t_1, t_2, ...)$, we are interested in a recursive scheme for the sequence of MMSE (minimum mean squared error) estimates of the target's kinematic state, given all available sensor observations. This requires a recursive computation of a conditional distribution. Like the EKF, our filter is based on the assumption that the conditional distribution of a target's kinematic state is well approximated by a Gaussian distribution. Note that, besides the usual sources of error in this approximation (sensor thresholds, nonlinear MTI measurements), the hybrid-state kinematics are nonlinear.

The filtering task is subdivided into the following tasks: *initialization*, *prediction* and *update*. We will make use of the following notation:

- $X(k \mid k)$ and $P(k \mid k)$ are the conditional mean and covariance, respectively, for the kinematic state of a target at time t_k, given all information up to and including time t_k
- $X(k+1 \mid k)$ and $P(k+1 \mid k)$ are the predicted mean and covariance, respectively, for the kinematic state of a target at time t_{k+1}, given all information up to and including time t_k
- $P(\cdot \mid \cdot) = \begin{bmatrix} P^{11}(\cdot \mid \cdot) & P^{12}(\cdot \mid \cdot) \\ P^{21}(\cdot \mid \cdot) & P^{22}(\cdot \mid \cdot) \end{bmatrix}$, where each of the matrices on the r.h.s. is a 2-by-2 matrix

The first report is used to initialize the filter. We assume there is no prior distribution associated with the target's position, and that the probability distribution for target velocity is given by $N(0, \Sigma_T)$. IMINT and SIGINT, we use a maximum likelihood estimate for the vehicle position.

In the case of MTI, the geometric projection of the report onto the ground is a good approximation to the maximum likelihood (ML) estimate. An accurate, iterative scheme to determine this projection makes use of the WGS84 ellipsoid model, DTED elevation data, and a correction for geoid undulation [4]. Alternatively, an approximate geometric projection is adequate in many settings [2]. This method neglects the difference in elevation of the initial target location relative to the origin of the Cartesian coordinate system, and provides a coarse approximation to the ML estimate of position. The range rate measurement is then processed in a standard manner, using the estimate of position to define the linearization point for the EKF.

Discretization of the continuous-time hybrid-state dynamics is based on the observation that the probability distribution for Markov-chain transitions is uniformly distributed on a time interval, given that a transition occurs during the time interval, and that transition delays are exponentially distributed [5]. This fact leads to closed-form prediction equations for the kinematic-state estimate and covariance matrix. Specifically, let $\Delta t_k = t_{k+1} - t_k$. If the discrete motion state is *move* at t_k and *stop* at t_{k+1}, we have:

$$X(k+1 \mid k) = \begin{bmatrix} x(k \mid k) + \frac{\Delta t_k}{2} \dot{x}(k \mid k) \\ y(k \mid k) + \frac{\Delta t_k}{2} \dot{y}(k \mid k) \\ 0 \\ 0 \end{bmatrix}, \tag{2.2}$$

$$
\begin{aligned}
P^{11}(k+1\mid k) &= P^{11}(k\mid k)+\frac{(\Delta t_k)^2}{3}P^{22}(k\mid k)\\
&\quad+\frac{\Delta t_k}{2}\left(P^{12}(k\mid k)+P^{21}(k\mid k)\right)\\
&\quad+\frac{(\Delta t_k)^2}{12}\begin{bmatrix}\dot{x}(k\mid k)\\ \dot{y}(k\mid k)\end{bmatrix}\begin{bmatrix}\dot{x}(k\mid k) & \dot{y}(k\mid k)\end{bmatrix}\\
&\quad+\begin{bmatrix}\frac{1}{12}q_x(\Delta t_k)^3 & 0\\ 0 & \frac{1}{12}q_y(\Delta t_k)^3\end{bmatrix},
\end{aligned}
\tag{2.3}
$$

$$
P^{12}(k+1\mid k)=P^{21}(k+1\mid k)=P^{22}(k+1\mid k)=0\,. \tag{2.4}
$$

If the discrete motion state is *stop* at t_k and *stop* at t_{k+1}, we have:

$$
X(k+1\mid k)=X(k\mid k), \tag{2.5}
$$

$$
P(k+1\mid k)=P(k\mid k). \tag{2.6}
$$

If the discrete motion state is *move* at t_k and *move* at t_{k+1}:

$$
X(k+1\mid k)=\Phi(\Delta t_k)X(k\mid k), \tag{2.7}
$$

$$
P(k+1\mid k)=\Phi(\Delta t_k)P(k\mid k)\Phi'(\Delta t_k)+Q(\Delta t_k), \tag{2.8}
$$

where $Q(\Delta t_k)$ is given by:

$$
Q(\Delta t_k)=\begin{bmatrix}
\frac{1}{3}q_x(\Delta t_k)^3 & 0 & \frac{1}{2}q_x(\Delta t_k)^2 & 0\\
0 & \frac{1}{3}q_y(\Delta t_k)^3 & 0 & \frac{1}{2}q_y(\Delta t_k)^2\\
\frac{1}{2}q_x(\Delta t_k)^2 & 0 & q_x\Delta t_k & 0\\
0 & \frac{1}{2}q_y(\Delta t_k)^2 & 0 & q_y\Delta t_k
\end{bmatrix} \tag{2.9}
$$

Finally, if the discrete motion state is *stop* at t_k and *move* at t_{k+1}:

$$
X(k+1\mid k)=X(k\mid k), \tag{2.10}
$$

$$
P(k+1\mid k)=\begin{bmatrix}
P^{11}(k\mid k)+\frac{(\Delta t_k)^2}{3}\Sigma_T & \frac{\Delta t_k}{2}\Sigma_T\\
\frac{\Delta t_k}{2}\Sigma_T & \Sigma_T
\end{bmatrix}+\hat{Q}(\Delta t_k), \tag{2.11}
$$

where $\hat{Q}(\Delta t_k)$ is given by :

$$
\hat{Q}(\Delta t_k)=\begin{bmatrix}
\frac{1}{12}q_x(\Delta t_k)^3 & 0 & \frac{1}{6}q_x(\Delta t_k)^2 & 0\\
0 & \frac{1}{12}q_y(\Delta t_k)^3 & 0 & \frac{1}{6}q_y(\Delta t_k)^2\\
\frac{1}{6}q_x(\Delta t_k)^2 & 0 & \frac{1}{2}q_x(\Delta t_k) & 0\\
0 & \frac{1}{6}q_y(\Delta t_k)^2 & 0 & \frac{1}{2}q_y(\Delta t_k)
\end{bmatrix}. \tag{2.12}
$$

If a sensor measurement is available, the prediction step is followed by an update step. This is a straightforward application of the EKF. The update equations depend on whether the report is MTI, IMINT, or SIGINT, and whether it corresponds to a mover or sitter. Note that an MTI sitter report corresponds to a stationary rotator, for which we only process range and azimuth components. Also, note that in processing MTI reports, available elevation data is used in determining the predicted range measurement.

There is a close connection between our hybrid-state model and filter, and the IMM approach [3]. Both approaches are characterized by continuous-state dynamics that are influenced by a time-varying discrete state. We note two differences between the approaches. First, in our model the discrete-state transitions are defined in continuous time: in general, the times associated with discrete-state transitions are not also times associated with sensor reports, and a discrete-time model does not reflect this. This affects both the prediction step of the filter, as well as the likelihood assigned to a particular discrete-state sequence. Secondly, our filter specifically accounts for discrete-state information provided by an upstream tracker.

3. IDENTITY MODELING AND FILTERING

A target's identity is inherently a static quantity. Nevertheless, it is useful to consider a dynamical model for vehicle identity: this introduces robustness into the resulting filtering equations, since the effect of process noise is to discount the weight of older identity observations relative to more recent ones. In particular, we characterize vehicle type by a finite-state Markov chain. Our knowledge of vehicle type is characterized by a probability distribution over all possible vehicle types. The distribution $v(k+1)$ corresponding to time t_{k+1} is computed recursively based on $v(k)$ and the current identity observation, $Z(k+1) \in \{1,...,m\}$. In particular, we have:

$$\hat{v}(k+1) = \left((1-\alpha) I + \frac{\alpha}{n} 1_n \right) v(k), \tag{3.1}$$

where I is the *n*-by-*n* identity matrix, 1_n is an *n*-by-*n* matrix of ones, and $\alpha \geq 0$. Note that by setting $\alpha = 0$, we have a static model for vehicle type. In general, we will use a small value for α.

If there is no identity observation, $v(k+1)$ is given by $\hat{v}(k+1)$. If there is an identity observation, $v(k+1)$ can be determined as follows:

$$v(k+1) = \frac{\hat{v}(k+1) N(j)}{\hat{v}(k+1) N(j) \bar{1}}, \tag{3.2}$$

where $\bar{1}$ is a vector of 1's of appropriate dimension, $N(j)$ is a diagonal matrix with $N(j)_{ii} = M_{ij}$, and M is the confusion matrices for the appropriate sensor.

The following quantity that represents the likelihood of a particular identity observation is used in our track scoring equations:

$$f(\hat{v}(k+1), Z(k+1)) = \begin{cases} \hat{v}(k+1) N(Z(k+1)) \bar{1}, & \text{if } Z(k+1) \text{ exists,} \\ 1, & \text{otherwise.} \end{cases} \tag{3.3}$$

4. TRACKER MODELS

The input to ATIF consists of the outputs of an arbitrary number of trackers. Currently, the following types of trackers are of interest:

- Moving-target-indicator (MTI) trackers
- Imagery-based (IMINT) trackers, including both EO and SAR
- Signal-intelligence (SIGINT) trackers

By assumption, each tracker produces a time-ordered sequence of *frames*. The contents of the frames are specified below. ATIF processes the merged sequence of frames from all the trackers in a time-ordered manner; the output is a sequence of frames indexed by the same times.

Each input frame consists of the following:

1. A *time stamp t*
2. A *sensor index*, which identifies the tracker
3. A *sensor type*, which identifies a sensor.
4. Possibly, a *bounding box* that characterizes the region on the ground that is illuminated by the sensor
5. A set of *reports*

If the bounding box is not specified, for MTI and IMINT frames we interpret this as meaning that the entire region of interest is included. Note that for IMINT frames, the sensor type will be either EO or SAR. SIGINT frames do not include a bounding box. Each report may include the following:

1. A track identifier, or *track id*
2. A *sensor observation*
3. A *discrete motion state*, which is either *move* or *stop*
4. An *identity observation*

We assume that every report includes a track id. If item (2) is missing from the report, we say that the track is *coasting*. A track is never initiated with a coast. For MTI, the sensor observation includes both the measured quantities as well as the platform position. For all the trackers currently of interest, items (3) and (4) are never included unless (2) is included. We do not have coasts associated with SIGINT tracks.

ATIF requires that each tracker have a characterization for its *detection* and *motion-state* assumptions. For the trackers that are currently of interest, these are shown in the following tables.

Tracker type	Detection of mover	Detection of sitter
MTI	1	0
IMINT (SAR frames)	0	1
IMINT (EO frames)	1	1
SIGINT	not applicable	not applicable

Table 4.1. Detection assumptions for specific sensors

Tracker type	Possible motion state	Default motion state (observation/coast)
MTI	move or stop	move/none
IMINT (SAR frames)	stop	stop/none
IMINT (EO frames)	move or stop	none/none
SIGINT	move or stop	none/not applicable

Table 4.2. Motion-state assumptions for reports from specific trackers

The information in these tables is relevant to the process by which ATIF generates track hypotheses. In Table 4.1, we indicate with a 1 those motion states that are detected with high probability by a particular

sensor, and we indicate with a 0 those motion states that are detected with low probability by a particular sensor. In particular, the lack of a report in a particular frame for an existing track hypothesis, combined with bounding box information and the information in Table 4.1, has implications for the motion state that is inferred for the target. If there is no report for a track for which one is expected, based on its previous estimated location relative to the bounding box and based on Table 4.1, the implication is that the (input) track has been terminated. Note that SIGINT tracks are never terminated.

Recall that a report may not include a discrete motion state. In Table 4.2, the second column indicates what are the possible discrete motion states. The third column indicates how ATIF will infer the discrete motion state when it is not specified. For MTI reports, the default assumption is that the sensor observation is associated with a moving target. Likewise, for SAR reports, the default assumption is that the sensor observation is associated with a stationary target. For EO and SIGINT, both discrete motion states are considered, and the likelier one is chosen. For all coasts, the likelier discrete motion state is chosen: this will correspond to maintaining the tracks' last discrete motion state.

While ATIF is currently based on inputs from the tracker types that have been described in this section, it is important to note that other tracker types may be added. To do so, it is necessary to identify the frame contents, as well as the detection and motion state assumptions.

5. HYPOTHESIS MANAGEMENT

The ATIF algorithm is a multi-hypothesis tracking (MHT) algorithm. That is, multiple association hypotheses are considered; these lead to a set of track hypothesis trees. With a specified latency, a single global hypothesis is chosen. A *global hypothesis* is a consistent set of track hypotheses that accounts for all incoming tracks in a consistent manner. In principle, all global hypotheses have a score that reflects their likelihood, based on the set of track inputs and our modeling assumptions. We identify a near-optimal global hypothesis, using a heuristic algorithm that does not require that all global hypotheses be explicitly enumerated. The identification of a global hypothesis is followed by a pruning scheme that removes all track hypotheses not contained in the global hypothesis that was selected. Note that, since the process of resolving to a single global hypothesis is done with some latency, multiple hypotheses are maintained.

ATIF follows the *track-oriented* paradigm introduced in [7], as opposed to the earlier, *hypothesis-oriented* paradigm originally proposed in [8]. (The latter approach has generated some renewed interest due to a more efficient implementation [9]). All tracks identified by an upstream tracker are accounted for at the ATIF output. The key difference between ATIF and report-level MHT tracking is that ATIF considers track-to-track association hypotheses, rather than report-to-report associations. While this reduces the set of association hypotheses that are considered for a given set of sensor reports, the algorithm must account for the widely varying update rates associated with upstream trackers. Unlike some track-to-track fusion algorithms [10][11], which combine state estimates obtained from different trackers, ATIF re-processes raw sensor measurements. This is done to enable improved state estimation as well as more accurate hypothesis scoring. Upstream trackers contribute to the fusion task by associating sensor measurements into tracks, and removing a substantial fraction of false alarms.

5.1. Hypothesis-Generation Logic

The track-generation logic identifies an updated set of track hypothesis trees, as a function of the existing set of track hypothesis trees and the current sensor frame. A high-level view of the logic is summarized in Table 5.1.

Each node (i.e. track hypothesis) in the set of track trees contains the following elements:

- time stamp
- set of (tracker index, track id) pairs
- discrete kinematic state: *move* or *stop*
- kinematic state: mean and covariance

- identity state
- score (based on track log-likelihood)

```
for i=1:number of active track hypotheses in previous layer
        if there is an update to the track and gating is successful
                add a track hypothesis to current layer
        else
                add a track hypothesis to current layer
                for all new tracks
                        if gating test is successful and association is allowed
                                add a track hypothesis to current layer
for i=1:number of reports
        if (sensor type, track id) pair is new or at least one gating test was unsuccessful
                add a track hypothesis to current layer, with no parent
```

Table 5.1. Pseudo-code for Track-Generation Logic

The *time stamp* is given by the current sensor frame time. The *set of (tracker index, track id) pairs* is the union of the following:

- For each MTI tracker, the (sensor index, track id) for the active MTI track associated with the node, if it exists.
- For each IMINT tracker, the (sensor index, track id) for the active IMINT track associated with the node, if it exists.
- For each SIGINT tracker, the union of all tracks associated with the node.

Also, if the parent's set of (sensor index, track id) pairs contains a pair for which an update exists in the current frame, but for which the gating condition is not satisfied, the node will have that (tracker index, track id) pair removed. Note that a track is not active if the latest frame associated with the particular tracker did not contain an update for the track.

Next, the *discrete kinematic state* is determined as follows. If a report in the current frame is used in generating a node, the report's discrete kinematic state defines that of the node, if such a discrete kinematic state exists. If it does not, both alternatives are considered, and the one resulting in a higher node *score* is chosen; a simple approximate to this task is to choose that hypothesis such that the discrete kinematic state does not change. This approximation is exact in the case of track coasts. Note that in general the same report may be used in different track hypotheses with different discrete motion states. This may occur with a coast report from any sensor type, and with any EO or SIGINT report for which no discrete motion state is specified. For new EO and SIGINT tracks where the first report's discrete motion state is not specified, by convention we always choose the *stop* state.

If a report is not used in generating the node, then in some instances a termination node is generated. Specifically, if the current sensor type is IMINT (EO) and the bounding box condition is satisfied, a termination node is generated. Also, for sufficiently unlikely *unconfirmed* mover or sitter tracks, i.e. track hypothesis that have transitioned from one discrete motion state to another without the presence of a sensor observation, a termination node is generated. Once a track has terminated, it cannot have any further reports associated with it.

In all other cases in which a report is not used in generating a node, the following logic determines the discrete kinematic state:

- If the current frame's sensor type is MTI and the bounding box condition is satisfied, the new node's discrete kinematic state is *stop*.
- If the current frame's sensor type is IMINT (SAR) and the bounding box condition is satisfied, the new node's discrete kinematic state is *move*.

- Otherwise, the new node's discrete kinematic state is the same as the parent node's discrete kinematic state.

The bounding box condition tests whether the location given by the latest kinematic state for a track hypothesis is contained in the region defined by the bounding box corner points.

The logic is motivated by two key ideas. The first is that the probability of detection associated with the MTI and SAR sensors are high for moving and stationary vehicles, respectively, when the vehicles are inside the sensor's bounding box. EO sensors have high probability of detection for both moving and stationary vehicles. This impacts the way that *negative information* is handled, i.e. the implicit information about a target that is available if no update is received in the current frame. In terms of the logic, a *coast*, i.e. a track update with no sensor observation, is handled in the same way as a track update with an observation. (Note that to avoid overshooting the stop location on MTI dropped tracks, a small latency allows terminal coasts to be removed). The second idea is that a discrete kinematic state transition is less likely than it is for a target to remain in the same discrete kinematic state. Thus, if no new information is available regarding a target's state, it is assumed that the discrete kinematic state has not changed.

Finally, note that the logic allows for upstream tracks to be broken. This is based on the recognition that errors in hypothesis pruning may occur. If at least one gating is unsuccessful for a track update, we spawn a new-track hypothesis. We have avoided additional hypotheses, whereby the report is also allowed to gate with existing tracks. In general, the gating parameters for track breakage will be different than for testing whether tracks may be fused, so that track fusion is more difficult than track breaks.

The gating test includes both kinematic and identity components, and insures that sufficiently unlikely hypotheses are not considered. Both kinematic and identity gating tests are performed; kinematic gating is based on the innovation likelihood associated with the EKF [12], while identity gating is based on eqn. (3.3).

5.2. Track Scoring

The log-likelihood score associated with a track initiation is given by

(5.1) $$l(1) = s,$$

where $s < 0$ is an ATIF parameter that penalizes track initiations. Note that there is no innovation likelihood term due to the first report: this reflects the fact that there is no prior distribution for target position. A track initiation due to an input track breakage is penalized more heavily still.

Let $l(k)$ be the score associated with a track. The general expression for updating the score is the following:

(5.2) $$\begin{aligned} l(k+1) &= l(k) \\ &\quad + \log\left[p_{dks(k),dks(k+1)}(t_{k+1}-t_k)\cdot g\left(X(k+1\mid k), P(k+1\mid k), Y_{k+1}\right)\cdot f\left(v(k), Z(k+1)\right)\right] \end{aligned}$$

where $p_{\cdot,\cdot}(\cdot)$ relates to discrete motion state transitions, $g(\cdot,\cdot,\cdot)$ is the innovation likelihood associated with the EKF, and $f(\cdot,\cdot)$ is given by eqn. (3.3) and relates to the identity observation.

Let $dks(k)$ denote the discrete kinematic state of a track at time t_k. The function $p_{\cdot,\cdot}(t_{k+1}-t_k)$ represents the probability of a single discrete state transition in the time interval (t_k, t_{k+1}), and is defined as follows for $\lambda_{sm} \neq \lambda_{ms}$:

(5.3) $$p_{mm}(t_{k+1}-t_k) = \exp\left(-\lambda_{ms}(t_{k+1}-t_k)\right),$$

(5.4) $$p_{ms}(t_{k+1}-t_k)=\frac{\lambda_{ms}}{(\lambda_{ms}-\lambda_{sm})}\left[\exp(-\lambda_{sm}(t_{k+1}-t_k))-\exp(-\lambda_{ms}(t_{k+1}-t_k))\right],$$

(5.5) $$p_{ss}(t_{k+1}-t_k)=\exp(-\lambda_{sm}(t_{k+1}-t_k)),$$

(5.6) $$p_{sm}(t_{k+1}-t_k)=\frac{\lambda_{sm}}{(\lambda_{ms}-\lambda_{sm})}\left[\exp(-\lambda_{sm}(t_{k+1}-t_k))-\exp(-\lambda_{ms}(t_{k+1}-t_k))\right].$$

For $\lambda_{sm}=\lambda_{ms}\equiv\lambda$, (5.4) and (5.6) are replaced by:

(5.7) $$p_{sm}(t_{k+1}-t_k)=p_{ms}(t_{k+1}-t_k)=\lambda(t_{k+1}-t_k)\left[\exp(-\lambda(t_{k+1}-t_k))\right].$$

The recursive expression (5.2) does not take account of normalization that goes on in ATIF. Once a track tree is *confirmed*, i.e. one of its branches is contained in all global hypotheses and an ATIF track id is assigned, the recursive likelihood computation is modified to account for the fact that, at the resolved layer, the track likelihood is set to 1. Intuitively, this sets to 1 the probability that a sequence of discrete state transitions and updates is associated with a particular target.

For a confirmed track tree, let *n* be the *N-scan* parameter. Once the track hypothesis at k+1-n is resolved, the normalized likelihood $\tilde{l}(k+1)$ differs from that defined in equation (5.2) as follows:

(5.10) $$\tilde{l}(k+1)=l(k+1)-l(k+1-n).$$

To clarify, the sequence of operations is as follows: when a new frame of data arrives, equation (5.2) is used to score each track hypothesis. These scores are used to determine which tracks are maintained and which are pruned, by reasoning over global hypotheses. Once pruning of the track trees has taken place, equation (5.9) is used to adjust the scores of all track hypotheses associated with confirmed track trees.

It is necessary to have a mechanism in place to remove confirmed tracks when no updates arrive for a sufficiently long period of time. The mechanism that we use is the following. When a track hypothesis first transitions from one discrete motion state to another without the presence of a sensor observation, we initialize a secondary likelihood calculation:

(5.12) $$l_2(k+1)=\log\left[p_{dks(k),dks(k+1)}(t_{k+1}-t_k)\right].$$

Subsequently, while no sensor observations are used in updating the track, the secondary likelihood is updated as follows:

(5.13) $$l_2(k+1)=l_2(k)+\log\left[p_{dks(k),dks(k+1)}(t_{k+1}-t_k)\right].$$

Note that normalization does not affect the secondary likelihood calculation.

A track termination may be due to a missing EO report, or may be due to an unconfirmed mover or sitter hypothesis with a sufficiently small secondary likelihood. For a track termination, the log-likelihood update equation is given by:

(5.14) $$l(k+1)=l(k)+d\ ,$$

where $d<0$ is an ATIF parameter that penalizes track terminations. Once a track has terminated, it cannot have any further reports associated with it, nor does its track score change.

5.3. Hypothesis Pruning

Our track-generation logic results in a set of track trees with a depth of *N_scan*+1, where *N_scan* defines the latency between the current frame of data and the resolved frame in ATIF. Further, scores are associated with each track hypothesis. In particular, each score is the log-likelihood associated with the track hypothesis, and depends on the discrete motion state sequence, kinematic state information, and vehicle type information, as well as additional algorithmic parameters including initiation, termination, and track-breakage scores (associated with breakage of input tracks).

Given a set of track trees, it is of interest to prune the trees so as to result in a depth of *N_scan*. That is, we wish to resolve all track hypotheses up to a latency of *N_scan*. To do so, one would like to identify the optimal global hypothesis from the set of all global hypotheses implicitly defined by the set of track hypotheses. A simple sub-optimal approach to doing so is to use a greedy heuristic, based on rank-ordering the track hypotheses [13]. This approach is not appropriate in the ATIF context, given the nature of our track hypothesis scores.

A possible approach is to use an indexing structure which explicitly identifies and updates a list of *K*-best global hypotheses, and resolves track hypotheses on the basis of the best global in the current list [9]. The disadvantage of this approach is that its complexity is quite large, i.e. roughly $O(KN^4)$, where *K* is the size of the global hypothesis list, and *N* is the number of targets in the scenario. Further, for a fixed level of performance, it appears that *K* must grow exponentially as a function of *N*. While elements of this approach may be of interest to us in the future, we have decided not to pursue it at present.

We have chosen to pursue a linear programming approach to identify a good approximation to the optimal global hypothesis. This approach identifies a good (near-optimal) global hypothesis, without requiring any global-hypothesis indexing structures.

Let *M* be the total number of input tracks to ATIF in the last *N_scan*+1 frames plus the total number of confirmed ATIF tracks, and let $\{t_i, i = 1,...M\}$ be the set of these tracks. Note that input tracks that are potentially broken by the ATIF track-generation logic are counted more than once, i.e. for the purposes of this discussion each track fragment is identified as a separate input track. Let $\{h_i, i = 1,...N\}$ denote the set of *N* ATIF track hypotheses. Let $\{\lambda_i, i = 1,...N\}$ denote the set of scores associated with the track hypotheses. Let $\{T_i, i = 1,...N\}$ denote the set of track lists associated with the track hypotheses. That is, the list $T_i \subset \{t_j, j = 1,...M\}$ identifies which input tracks are contained in the ATIF track hypothesis h_i.

A global hypothesis is identified by a vector $x \in \{0,1\}^N$ where, denoting as x_i the *i*th component of *x*, we interpret $x_i = 1$ to mean that hypothesis h_i is contained in the global hypothesis. A global hypothesis must satisfy

(5.15) $Ax = b$,

where *A* is an *M*x*N* matrix with $A_{ij} = 1$ if $t_i \in T_j$, i.e. if the input track t_i is part of track hypothesis h_j. Otherwise, we have $A_{ij} = 0$. Also, *b* is $\{1\}^M$, i.e. a vector of ones.

The constraint (2.1) says that a global hypothesis must account for all input tracks, and must not include the same input track in more than one track hypothesis. The optimal global hypothesis $\hat{x}$ is given by

(5.16) $\hat{x} = \arg\max c'x$,

subject to (2.1) and $x \in \{0,1\}^N$, where c is a vector with $c_i = \lambda_i, i = 1,...N$. Let us denote this optimization problem by P1. Solving P1 directly for $\hat{x}$ is an intractable, integer programming problem.

If we relax the constraint $x \in \{0,1\}^N$ to $x \in [0,1]^N$, we have a standard linear programming problem with both equality and inequality constraints. Let us denote this problem as P2, and let $\tilde{x}$ denote the solution to P2. Note that in general we will have $\tilde{x} \notin \{0,1\}^N$.

In order to obtain a feasible solution to P1, it is necessary to perturb the solution $\tilde{x}$ to P2. A simple way to do so is the following. Consider a third optimization problem P3:

(5.17) $\max \tilde{x}'x$,

subject to (2.1) and $x \in \{0,1\}^N$. Let $\bar{x}$ denote the sub-optimal solution to P3 obtained using the greedy heuristic. This is our approximation to $\hat{x}$.

Our use of the greedy heuristic applied to the same set of track hypotheses with track scores given by the solution to P2, amounts to a sequential rounding off to 1 of elements of $\tilde{x}$, done in a greedy fashion beginning with those elements close to 1. Using the greedy heuristic insures that a feasible solution vector ($\bar{x}$) results.

6. PERFORMANCE RESULTS

For both of the scenarios that we will consider, ground truth target trajectories and vehicle type information are available. For performance studies, one could consider simulating sensor reports and providing these reports to single-sensor trackers, and then providing the outputs to ATIF. While this approach is appropriate to study overall system performance, it does not isolate the performance of the ATIF algorithm. Thus, we have developed track-simulation code that generates MTI, IMINT, and SIGINT tracks on the basic of ground truth data and a number of track-simulation parameters. This allows us to treat the quality of upstream tracks as an input specification.

Input tracks are simulated as follows. Noisy report level information is generated with zero-mean white Gaussian sensor noise, for all sensor types. Identity information is confused using confusion matrices that are representative of typical performance. In the case of MTI, identity information is coarse, i.e. vehicles are classified as *tracked* or *wheeled*. In the case of IMINT, based on existing model-based classification (MBC) technology, confusion matrices are available which are diagonally dominant, though some vehicle types cannot be distinguished. For SIGINT, we use a diagonally dominant square confusion matrix. We simulate missing sensor reports (coasts) and missing identity information. All MTI reports are currently associated with moving targets (no simulation of stationary rotators), and all SAR reports are associated with stationary targets. EO and SIGINT reports may or may not contain discrete state information.

Track breaks are introduced at each track update, with some probability; also, all MTI tracks are broken when targets are stationary, and all IMINT tracks are broken when a sufficiently large displacement occurs. Sensor frame times may be defined by specifying collection rates for each sensor; alternatively, specific collection times and bounding box information may be specified. The latter allows us to simulate tracks consistent with specific data collections associated with a scenario of interest.

The key ATIF performance metrics that we focus on are the following:

- *Fragmentation* (FRAG). This is defined as the average number of ATIF tracks per target. Good performance corresponds to low fragmentation.

- *Probability of correct track-to-track association* (PCA). This is computed as the ratio of the number of correct associations and the total number of associations.

- *Group-level probability of correct track-to-track association* (GPCA). This is a slight modification to the PCA metric, whereby we do not distinguish between targets in the same group. Groups are defined on a scenario-by-scenario basis.

- *Probability of track breakage* (PTB). As discussed, ATIF will occasionally break incoming tracks, principally for those trackers that have a low update rate relative to other trackers, and when N_scan is not sufficiently large. PTB is computed as the ratio of the number of track breaks and the total number of input tracks.

The performance of ATIF depends on the scenario, the quality of the incoming tracks, and a number of key algorithmic parameters. Extensive Monte Carlo performance testing is needed to fully characterize the dependence of our performance metrics on these. At present, our testing has proceeded as follows. For each of the two scenarios that we consider, we generate input tracks of varying quality, where the lower-quality tracks are characterized by much higher fragmentation. We run ATIF in a *high-fusion* or *low-fusion* mode, where the high-fusion mode considers many more association-hypotheses be using much larger kinematic and identity gates. Finally, ATIF is run in a *delayed-resolution* (MHT) or *immediate-resolution* (non-MHT) mode, where the first is characterized by a large enough N_scan to allow the update of low update rate tracks, particularly IMINT tracks, while the second has N_scan=0.

Fig. 6.1 illustrates the qualitative behavior that we expect as a function of the quality of input tracks and key ATIF parameters. In particular, the lines illustrate the tradeoff between high PCA and low fragmentation: for a given set of input tracks and a fixed N_scan, as ATIF more aggressively fuses input tracks, PCA drops off. Conversely, a high PCA corresponds to a conservative, low-fusion mode. Both metrics improve as the quality of the input tracks improves, or as the complexity of the algorithm is increased with a larger accrual of information before hypotheses are resolved, i.e. large N_scan. A large N_scan allows ATIF to more aggressively create association hypotheses, without decreasing the resulting PCA.

A similar qualitative behavior is expected if PCA is replaced by GPCA. Note that, in addition to a lower PCA, the high fusion mode coupled with a low N_scan is more likely to induce some input track breaks.

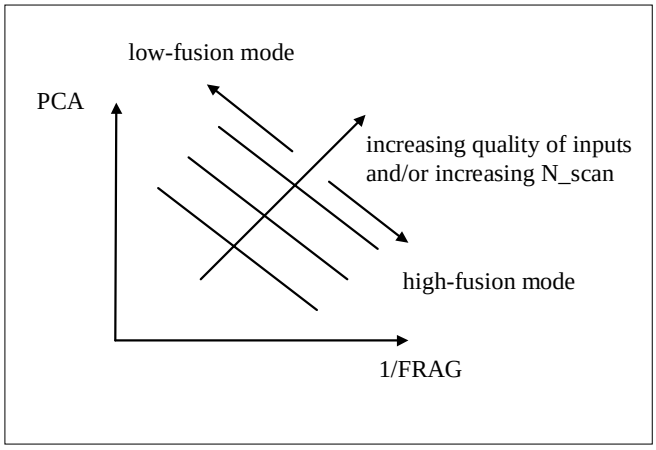

Figure 6.1. ATIF performance tradeoffs

6.1. First Scenario

The first scenario is comprised of 30 targets, and takes place over a one-hour period. Of the 30 targets, 19 are stationary targets that only give rise to IMINT or SIGINT tracks. 13 of these are targets at Site 12 that are observed in imagery; we define these to be a group. There are 11 targets that go through a number of *stop-move* and *move-stop* transitions. A first group of three targets is imaged at Site A and subsequently travel to Site C. The second and third groups (three and five targets, respectively) are imaged at Site B

along with the 13 “clutter” targets. Subsequently, seven of the eight targets in groups two and three move a short distance towards Site C, and stop. A second image of Site B is taken, without these seven targets. At this point, group two moves to the water (Site C), and stops. An image at Site C is taken, containing the group two targets. Group 3 moves from near Site B to Site C, moves across the water, and proceeds northwest. Finally, groups one and two move together back to Site B. A number of SIGINT reports (both COMINT and ELINT) are interspersed throughout the one-hour scenario, originating from the three groups of moving targets as well as from six additional fixed targets. Fig. 6.2 illustrates the target trajectories associated with this scenario, as well as the bounding boxes of the SAR images.

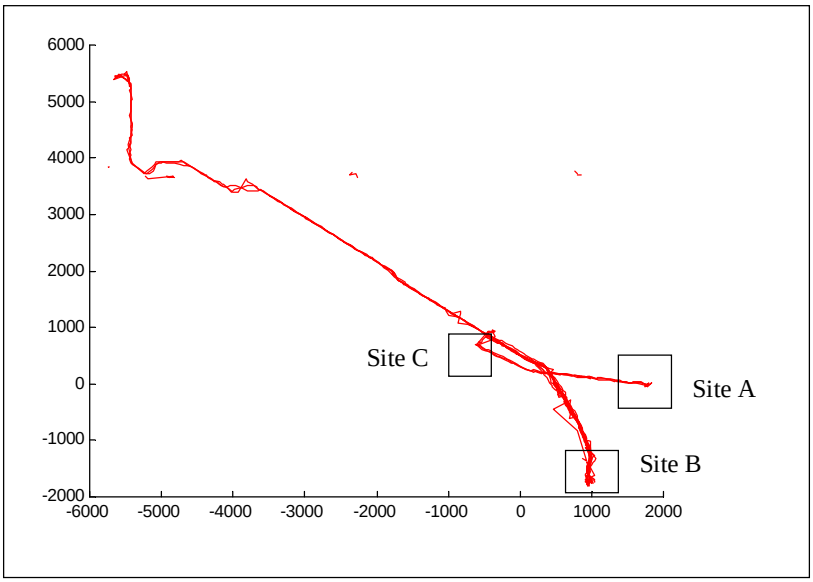

Figure 6.2. The First Scenario

Monte Carlo based performance results are illustrated in Table 6.1. In particular, note that we have considered all combinations of *high-fusion* and *low-fusion* modes, with MHT and non-MHT modes. As expected, larger fusion gates lead to improved fragmentation, at the cost of decreased PCA and GPCA. The delayed-resolution (MHT) mode improves performance with respect to all metrics.

ATIF mode	FRAG	PCA	GPCA	PTB
high fusion, non-MHT	1.0933	0.4213	0.6820	0.0186
low fusion, non-MHT	2.1467	0.4444	0.6897	0.0153
high fusion, MHT	1.0200	0.4751	0.7843	0.0051
low fusion, MHT	2.1200	0.4729	0.7493	0.0119

Table 6.1. Performance Metrics for the First Scenario

6.2. Second Scenario

Unlike the first scenario, the second scenario is based on simulated ground truth tracks. These were developed to provide a scenario of current military interest, where a set of targets scatters as an attack is underway. The scenario is comprised of 8 targets, partitioned into four groups of size 2, 3, 2, 1,

respectively. Groups 1 and 4 move from Site A to Site C, after which imagery is collected at Site C. Groups 2 and 3 similarly proceed from Site A to Site C (and the Group 4 target repositions itself slightly), followed by imagery at Site C over all 8 targets. Sensing a strike, the assembled vehicles depart in stages. Group 2 moves to Site D and group 1 moves to Site B. Imagery is again collected over Site C, as well as over Site B. Finally, the remaining groups leave Site C, Site C is imaged after all vehicles have departed, Groups 3 and 4 arrive at Site B, and images are taken at Sites B and D. SIGINT reports exist over the early part of the scenario, before the targets are of aware of an incoming strike. Figure 6.3 illustrates the target trajectories associated with this scenario, as well as the bounding boxes of the SAR images.

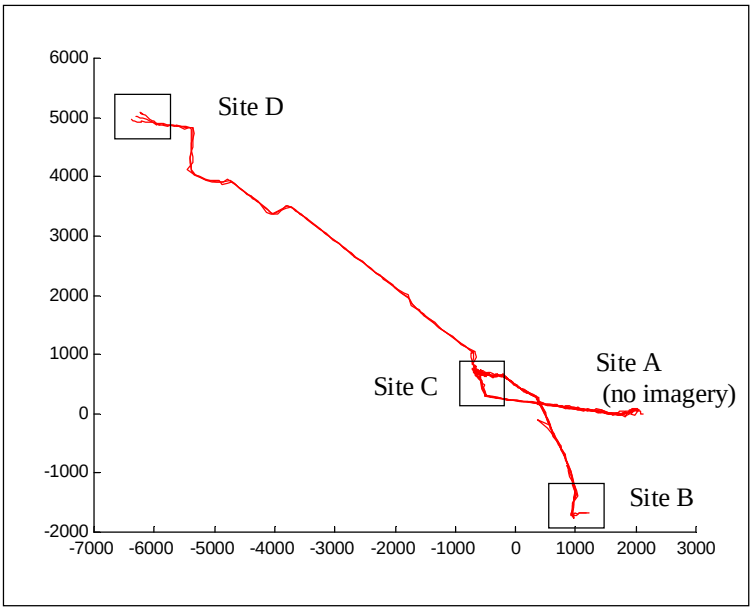

Figure 6.3. The Second Scenario

Monte Carlo based performance results are illustrated in Table 6.2, for the same ATIF modes considered in the river-crossing scenario. Note that performance results are better for this scenario; this is due in part to the improved, noiseless ground truth data that led to better quality in the simulated input tracks.

ATIF mode	FRAG	PCA	GPCA	PTB
high fusion, non-MHT	1.0250	0.7571	0.8500	0.0056
low fusion, non-MHT	1.4750	0.8350	0.9345	0.0056
high fusion, MHT	1.0250	0.8627	0.9780	0
low fusion, MHT	1.4750	0.8671	0.9837	0

Table 6.2. Performance Metrics for the Second Scenario

Interestingly, in the MHT mode, there is a smaller drop-off in PCA and GPCA as we go from the low-fusion to the high-fusion mode, as compared with the drop-off in the non-MHT mode. (In fact, in the water-movement scenario, there is no drop-off at all). This suggests that, when using delayed hypothesis resolution, we need not be conservative in considering potential association hypotheses, since in the high-fusion mode we have a considerable reduction in fragmentation with only a small penalty, if any, in PCA and GPCA.

Finally, note that performance results depend on the quality of the input tracks. Figure 6.4 illustrates that ATIF fragmentation increases as the number in input tracks is increased, due to higher fragmentation in the upstream trackers.

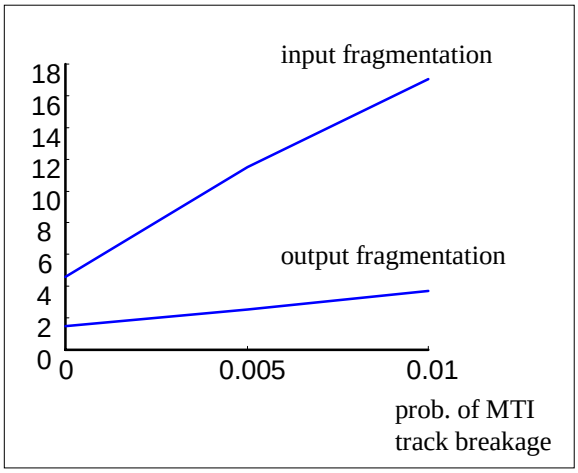

Figure 6.4. Input and Output Fragmentation for the Second Scenario

7. CONCLUDING REMARKS

This paper has introduced a new track-to-track association algorithm (ATIF) that fuses tracks from an arbitrary numbers of MTI, IMINT (both SAR and EO), and SIGINT trackers. The ATIF algorithm considers multiple association hypotheses and uses evidence accrual to resolve hypotheses with a specified latency. Track hypothesis scores are based on track information, kinematic information, and vehicle type information. The algorithm effectively fuses information from disparate sources to provide better tracking accuracy, vehicle type information, and track continuity, with a good probability of correct association (especially at the group level). A key feature of the ATIF algorithm is that it is readily extensible to additional sensor/tracker types, provided that appropriate models for these are available, e.g. IR imagery.

Preliminary analysis of the complexity of the ATIF algorithm suggests that it should scale well with increasing scenario size. We plan to validate this analysis by testing the algorithm on larger scenarios. Also, we plan to explore extending the algorithm to include road tracking, tracking at a group level, adaptive hypothesis filtering, and asynchronous tracking and filtering issues that arise in large, decentralized networks of trackers. Finally, we plan to develop a closed-loop architecture whereby sensor management can improve the results provided by the current, open-loop architecture.

REFERENCES

1. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.
2. S. Coraluppi, M. Luettgen, and C. Carthel, A Hybrid-State Estimation Algorithms for Multi-Sensor Target Tracking, in *Proc. 3rd International Conference on Information Fusion*, July 2000, Paris, France.
3. Mazor, Averbuch, Bar-Shalom, and Dayan, Interacting Multiple Model Methods in Target Tracking: A Survey, *IEEE Transactions on Aerospace and Electronic Systems*, 34(1), 1998.
4. M. Mallick, Maximum Likelihood Estimation of Target Geo-location Using Range-Azimuth Measurements, Terrain Elevation, and Geoid Undulation, February 2000.
5. H. Tijms, *Stochastic Models*, John Wiley & Sons, 1994.
6. G. Strang and K. Borre, *Linear Algebra, Geodesy, and GPS*, Wellesley-Cambridge Press, 1997.
7. T. Kurien, Issues in the Design of Practical Multitarget Tracking Algorithms, in *Multitarget-Multisensor Tracking*, Y. Bar-Shalom (ed.), pp. 43-83, Artech House, 1990.

8. D. Reid, An Algorithm for Tracking Multiple Targets, *IEEE Transactions on Automatic Control 24(6)*, pp. 843-854, 1979.
9. I. Cox and M. Miller, On Finding Ranked Assignments with Application in Multitarget Tracking and Motion Correspondence, *IEEE Transactions on Aerospace and Electronic Systems* 31(1), pp. 486-489, 1995.
10. R. Lobbia and M. Kent, Data Fusion of Decentralized Local Tracker Outputs, *IEEE Transactions on Aerospace and Electronic Systems* 30(3), pp. 787-799, 1994.
11. K.C. Chang, R.K. Saha, and Y. Bar-Shalom, On Optimal Track-to-Track Fusion, *IEEE Transactions on Aerospace and Electronic Systems* 33(4), October 1997.
12. P.R. Kumar and P. Varaiya, Stochastic Systems: Estimation, Identification, and Adaptive Control, Prentice-Hamm Inc., 1986.
13. TRW System Integration Group, Algorithm Survey Report Preliminary Review Copy, Doc. No. 52609.121-004, November 1992.